Project Background

Smart Sensor Interface

Design a smart sensor interface that includes analog circuits and utilizes machine learning algorithms to analyze sensor data in real-time. This interface can be used in various applications such as environmental monitoring and physiological data analysis in medical devices.

Sensor Application Fields: Environmental Monitoring

Example:

- Air Quality Monitoring (PM2.5): Real-time collection of pollutant concentration data in the air to provide scientific basis for environmental protection and management.

- Water Quality Detection: Real-time detection of pH value, dissolved oxygen, heavy metals, and other parameters in water using the smart sensor interface. This helps in promptly identifying water quality issues and protecting water resources.

- Meteorological Monitoring: Integration of temperature, humidity, wind speed, and precipitation sensors. The smart sensor interface can monitor weather changes in real-time, providing accurate weather forecasts and disaster warnings.

Another Application Layer for Sensors: Medical Devices for Vital Sign Monitoring

Examples:

- Wearable Devices: Smart sensors can collect physiological data such as heart rate, blood oxygen, and body temperature in real-time. Machine learning algorithms can analyze data changes, enabling early detection of health problems like heart disease and diabetes.

- Remote Healthcare: Through the smart sensor interface, doctors can remotely obtain patients' real-time physiological data for diagnosis and monitoring, facilitating patient care.

Project Workflow Introduction

First, we need to select suitable types of sensors. Choosing the right sensors based on application scenarios is crucial.

1. Sensor Selection and Interface Design

a. Sensor Selection:

- Temperature Sensors: e.g., NTC thermistors, RTD
- Humidity Sensors: e.g., capacitive humidity sensors

b. Analog Circuit Design:

Amplifier circuits, filter circuits, and A/D conversion (converting analog signals to digital signals)

2. Data Acquisition and Preprocessing (Embedded System)

The conditioned sensor signals are read and transmitted to the data processing unit through microelectronic components, such as operational amplifiers and analog-to-digital converters.

- First, the sensor signals go through front-end conditioning circuits, including amplification, filtering, and calibration to ensure stability and accuracy.
- Next, ADCs convert the analog signals to digital signals.
- Then, these digital signals are read by microcontrollers (MCUs) or microprocessors (MPUs). MCUs or MPUs perform initial processing and storage and transmit data to the data processing unit through wired or wireless communication interfaces (e.g., UART, I2C, SPI, Wi-Fi, or Bluetooth).
- In the data processing unit, further analysis and processing of signals are performed to achieve real-time monitoring and control functions. This ensures high precision and reliability from sensor data acquisition to processing.

For this project, we will use an Arduino as the microcontroller, the Arduino UNO R4 model with a Wi-Fi module.

The purpose of collecting data after designing the sensor: Calibration bias to make data collection more accurate is one part. The most important is to test the effectiveness of our sensor design. For instance, if the current temperature is 25°C and the sensor reads 5°C, it indicates a major design issue, and data collection helps to verify the sensor design.

3. Application of Machine Learning

In the application of machine learning, we will use a linear regression model to process data by applying common statistical methods such as mean, variance, and standard deviation, and then analyze and process sensor data in real-time to provide predictive results.